








Reto 5: Distribución de vacunas - Corporación Umbrella

 Propósito	Generar 3 reportes en los cuales se hará uso de listas de tuplas, y operaciones entre estas.
 Dificultad	
 Fecha	@May 20, 2022
 Fuente	MisionTic2022 Ciclo 1
 Numero	24
 Adjuntos	Reto_Semana_5 - 16mayo2022.pdf

Autor: **David Santiago Urbano Rivadeneira**

Proceso Ideal

1. **Identificar** el problema

- ¿Cual es el problema? → El Departamento de Logística de Corporación Umbrella ha quedado muy contento con tu desarrollo anterior, puesto que ahora puede administrar los despachos de las vacunas con mayor eficiencia. Por lo tanto a decidido aumentar la capacidad de sus camiones con 1000 cajas.

Debido al posible robo de mercancía, se es necesario generar un reporte donde se comparen planillas de lo registrado por los conductores encargados de repartir las vacunas en los puntos de distribucion y lo asignado por la corporacion Umbrella.

- Stakeholders

- Corporación Umbrella (cliente y usuario)
- Conductores de Transportes del Norte. (usuarios)



Nota: Cada uno de los usuarios ingresara planillas (datos de entrada) al programa

- Restricciones

Restricciones

Habra como maximo 10 puntos de distribucion por Conductor

2. Definir el problema

- ¿Que conozco? La planilla que se ingresara al programa consta de 3 variables (**Punto de distribucion**, **Cajas entregadas** y **Tiempo de despacho**).
- Se dividio el problema en subproblemas
 - Realizar la resta o diferencia de **cajas entregadas** y **tiempo de despacho**
 - Imprimir 3 puntos de distribución que registran mayores sobre-entregas.
 - Imprimir los 3 puntos de distribución que registran las mayores demoras.
 - Imprimir todos los puntos de distribución que registran ambas condiciones negativas: tanto sobre-entregas, como demora en su tiempo de despacho.

3. Estrategia

Ejemplos:

Entrada del programa:

```
lista_tuplas_asignado = [(1, 98, 11), (2, 86, 14), (3, 99, 11), (4, 89, 12), (
    5, 89, 12), (6, 96, 10), (7, 93, 13), (8, 87, 15), (9, 89, 10), (10, 92, 10)]

lista_tuplas_registrado = [(1, 100, 10), (2, 86, 10), (3, 97, 15), (4, 93, 15), (
    5, 94, 12), (6, 93, 13), (7, 95, 12), (8, 85, 11), (9, 90, 11), (10, 90, 12)]
```

Salida esperada del programa:

```
Punto # 1

Diferencia de cajas = -2

Diferencia de tiempos = 1

Eficiencia = 9.1%

Punto # 2

Diferencia de cajas = 0

Diferencia de tiempos = 4

Eficiencia = 28.6%

. . . (Asi mismo para cada punto hasta el 10)

Punto # 10

Diferencia de cajas = 2

Diferencia de tiempos = -2

Eficiencia = -20.0%

Puntos con mayores demoras de tiempo:

Punto 3: -4

Punto 4: -3

Punto 6: -3

Puntos con mayores sobre-entrega:

Punto 5: -5

Punto 4: -4

Punto 1: -2

Puntos con los dos criterios negativos:

Punto 4

Punto 9
```

4. Algoritmos

Requerimientos funcionales

El aplicativo debera calcular la diferencia de cajas entregadas, reflejando “lo asignado” menos “lo registrado”. Los valores negativos corresponden a sobre-entregas (unidades entregadas de más).

El aplicativo debera calcular la diferencia de tiempos de despacho, reflejando “lo asignado” menos “lo registrado”. Los valores negativos corresponden a demoras en el punto de distribución.

El aplicativo debera calcular la eficiencia en los tiempos de despacho, reflejada como un porcentaje que toma como referencia “lo asignado”.

- Algoritmo diferencia de lista tuplas

Las matrices son estructuras de datos que tienen un uso muy importante en las matemáticas, en Python las matrices se representan como listas de listas, la resta y suma de matrices solo se puede efectuar cuando son de la misma dimension como en este caso. nuestra matriz de 10*3, de esta manera si representamos a la matriz resultante como c, se tiene que:

$$c_{ij} = a_{ij} - b_{ij}$$

Pseudocodigo

Nombre: Resta de matrices

“Este algoritmo realiza la resta entre dos matrices y el resultado es almacenado en otra matriz”

INICIO

Variables:

Lista de tuplas: lista_tuplas_asignado, lista_tuplas_registrado

Recibir: lista_tuplas_asignado, lista_tuplas_registrado

```
c ← []
```

```

Para k ← 0 hasta k < longitud(lista_tuplas_asignado):
    Para k ← 0 hasta j < longitud(lista_tuplas_asignado[0]):
        c[k][j] = lista_tuplas_asignado[k][j] - lista_tuplas_registro[k][j]

```

FIN

- Algoritmo diferencia de lista tuplas



El anterior algoritmo realizaba la resta entre las dos matrices y al hacerlo borraba el identificador de los puntos de distribución, por lo que se deben agregar de nuevo, porque tenemos que imprimirlos mas adelante

Nombre: Suma de matrices

“””Este algoritmo realiza la suma entre dos matrices“””

INICIO

Variables:

Lista de tuplas: c, enumeracion

Recibir: lista_tuplas_asignado, c

```

enumeracion ← [(1, 0, 0), (2, 0, 0), (3, 0, 0), (4, 0, 0), (5, 0, 0), (6, 0, 0), (7,
0, 0), (8, 0, 0), (9, 0, 0), (10, 0, 0)]
Para k ← 0 hasta k < longitud(lista_tuplas_asignado):
    Para k ← 0 hasta j < longitud(lista_tuplas_asignado[0]):
        c[k][j] <- enumeracion[k][j] + c[k][j]

```

FIN

Codigo de la solucion → [Link al replit](#)



Siguiendo las **convenciones estilísticas** de la PEP8 para escribir código en Python mas legible

Código Main.py

```
import control

lista_tuplas_asignado = [
    (1, 98, 11),
    (2, 86, 14),
    (3, 99, 11),
    (4, 89, 12),
    (5, 89, 12),
    (6, 96, 10),
    (7, 93, 13),
    (8, 87, 15),
    (9, 89, 10),
    (10, 92, 10),
]

lista_tuplas_registrado = [
    (1, 100, 10),
    (2, 86, 10),
    (3, 97, 15),
    (4, 93, 15),
    (5, 94, 12),
    (6, 93, 13),
    (7, 95, 12),
    (8, 85, 11),
    (9, 90, 11),
    (10, 90, 12),
]

control.estadistica(lista_tuplas_asignado, lista_tuplas_registrado)
```

Control.py

```

def estadistica(lista_tuplas_asignado, lista_tuplas_registrado):
    """Imprime un reporte que contiene:
    - Los 3 puntos de distribución que registran mayores sobre-entregas.
    - Los 3 puntos de distribución que registran las mayores demoras.
    - Todos los puntos de distribución que registran ambas condiciones negativas: tanto sobre-entregas,
    como demora en su tiempo de despacho.

    Args:
        lista_tuplas_asignado (list)
        lista_tuplas_registrado (list)

    Returns:
        none
    """
    a = lista_tuplas_asignado
    b = lista_tuplas_registrado

    # Matriz resultado 10 filas * 3 columnas
    c = [[None for columnas in range(0, 3)] for filas in range(0, 10)]

    no_filas = len(a)
    no_columnas = len(a[1])

    for k in range(no_filas): # Itera las filas.
        for j in range(no_columnas): # Itera las columnas.
            c[k][j] = a[k][j] - b[k][j] # Realiza la resta.

    # Añadimos de nuevo la enumeracion de los puntos
    enumeracion = [(1, 0, 0), (2, 0, 0), (3, 0, 0), (4, 0, 0), (5, 0, 0),
                    (6, 0, 0), (7, 0, 0), (8, 0, 0), (9, 0, 0), (10, 0, 0)]
    for k in range(no_filas):
        for j in range(no_columnas):
            c[k][j] += enumeracion[k][j]

    for i in range(len(a)):
        print(f"Punto #{i+1}\n") # aqui tambien podria usar el indice
        print("Diferencia de cajas = ", c[i][1])
        print("Diferencia de tiempos =", c[i][2])
        eficiencia = c[i][2] / a[i][2]
        print(f"Eficiencia = {eficiencia:.1%}\n")

    # Puntos con mayores demoras de tiempo
    c.sort(key=lambda x: x[2]) # Aplicamos ordenamiento
    print("Puntos con mayores demoras de tiempo:\n")
    for i in range(3):
        print(f"Punto {c[i][0]}: {c[i][2]}\n")

    # Puntos con mayores sobre-entrega
    c.sort(key=lambda x: x[1])
    print("Puntos con mayores sobre-entrega:\n")
    for i in range(3):
        print(f"Punto {c[i][0]}: {c[i][1]}\n")

    # Puntos con los dos criterios negativos
    print("Puntos con los dos criterios negativos\n")
    for i in range(len(c)):
        if c[i][2] < 0 and c[i][1] < 0:
            print(f"Punto {c[i][0]}\n")

```

Pruebas

The screenshot shows a code editor with a file named `main.py`. The code defines two lists of tuples, `lista_tuplas_asignado` and `lista_tuplas_registro`, and calls a function `control.estadistica` with these lists as arguments. The output in the console shows the results for nine points, each with three metrics: `Diferencia de cajas`, `Diferencia de tiempos`, and `Eficiencia`.

```
1 import control
2
3
4 lista_tuplas_asignado = [
5     (1, 98, 11),
6     (2, 86, 14),
7     (3, 99, 11),
8     (4, 89, 12),
9     (5, 89, 12),
10    (6, 96, 10),
11    (7, 93, 13),
12    (8, 87, 15),
13    (9, 89, 10),
14    (10, 92, 10),
15 ]
16
17 lista_tuplas_registro = [
18     (1, 100, 10),
19     (2, 86, 10),
20     (3, 97, 15),
21     (4, 93, 15),
22     (5, 94, 12),
23     (6, 93, 13),
24     (7, 95, 12),
25     (8, 85, 11),
26     (9, 90, 11),
27     (10, 90, 12),
28 ]
29
30 control.estadistica(lista_tuplas_asignado,
31                     lista_tuplas_registro)
```

Console Output:

```
Punto #1
Diferencia de cajas = -2
Diferencia de tiempos = 1
Eficiencia = 9.1%

Punto #2
Diferencia de cajas = 0
Diferencia de tiempos = 4
Eficiencia = 28.6%

Punto #3
Diferencia de cajas = 2
Diferencia de tiempos = -4
Eficiencia = -36.4%

Punto #4
Diferencia de cajas = -4
Diferencia de tiempos = -3
Eficiencia = -25.0%

Punto #5
Diferencia de cajas = -5
Diferencia de tiempos = 0
Eficiencia = 0.0%

Punto #6
Diferencia de cajas = 3
Diferencia de tiempos = -3
Eficiencia = -30.0%

Punto #7
Diferencia de cajas = -2
Diferencia de tiempos = 1
Eficiencia = 7.7%

Punto #8
Diferencia de cajas = 2
Diferencia de tiempos = 4
Eficiencia = 26.7%

Punto #9
Diferencia de cajas = -1
Diferencia de tiempos = -1
Eficiencia = -10.0%
```



Cambie algunas cosas en el código para pasar las pruebas

Punto #10

Diferencia de cajas = 2
Diferencia de tiempos = -2
Eficiencia = -20.0%

Puntos con mayores demoras de tiempo:

Punto 3: -4

Punto 4: -3

Punto 6: -3

Puntos con mayores sobre-entrega:

Punto 5: -5

Punto 4: -4


Punto 1: -2

Puntos con los dos criterios negativos

Punto 4

Punto 9



 [Link notion](#)